Electronic Version
Stylesheet Version v1.1.1

Description

METHOD AND APPARATUS FOR MEMORY ALLOCATION

BACKGROUND OF INVENTION

[0001] FIELD OF THE INVENTION

[0002] The present invention relates generally to integrated circuits, and more particularly to methods and apparatus for memory allocation within integrated circuits.

[0003] BACKGROUND

[0004] An integrated circuit (IC), such as a system on a chip (SOC), may include a processing unit (e.g., a processor) which includes a subsystem (e.g., an internal memory). The processor may access the subsystem to execute an instruction or read and write data.

[0005] The IC may include one or more direct memory access (DMA) machines coupled to the processor. The DMA machines may also access the subsystem of the processor to load (e.g., preload) code or data into the subsystem be-

fore the processor executes an instruction which requires the code or data.

[0006] An application running on the IC may have a time budget that requires one or more of the DMA machines to load code and/or data into the subsystem, and the processor to execute instructions within allotted times, respectively. If one or more of the DMA machines does not load (e.g., preload) code or data, which is required by instructions to be executed by the processor, into the subsystem within the allotted time, the IC fails and the application is non-operative. Similarly, if the processor does not execute an instruction within the allotted time, the IC fails and the application is non-operative.

[0007] According to one scheme, a fixed priority may be assigned to the processor and one or more of the DMA machines. A task received from the highest-priority processor or DMA machine may be completed before other tasks. If all DMA machines are assigned a higher priority than the processor, the tasks of all the DMA machines are completed before any processor tasks. In such a system, a processor task will eventually fail to be completed within an allotted time. Alternatively, if the processor is assigned a higher priority than the DMA machines, all the processor

tasks are always completed before any tasks of the DMA machines. In such a system, one or more DMA machines tasks will eventually fail to be completed within the allotted time. For example, one or more of the DMA machines may fail to load data into the subsystem of the processor before execution of the instructions that require the data.

[0008] Because the processor and one or more DMA machines must access the subsystem to complete tasks, respectively, within the allotted times, a method of allocating priority between the processor and the one or more DMA machines is needed.

Summary of Invention

[0009] In a first aspect of the invention, a first method is provided for allocating memory bandwidth. The first method includes the steps of (1) assigning a fixed priority of access to memory bandwidth to one or more direct memory access (DMA) machines; and (2) assigning a programmable priority of access to memory bandwidth to a processing unit. The programmable priority of the processing unit allows priority allocation between the one or more DMA machines and the processing unit to be adjusted dynamically.

[0010] In a second aspect of the invention, a first apparatus is

provided that includes (1) a processing unit for executing tasks; (2) one or more direct memory access (DMA) machines each responsible for retrieving data and providing the retrieved data to the processing unit; (3) a bus, coupled to the processing unit and the one or more DMA machines, for providing communication between each of the one or more DMA machines, the processing unit and a data resource; and (4) a dynamic priority allocation circuit for allocating priority of access to the data resource between each of the one or more DMA machines and processing unit. The dynamic priority allocation circuit is adapted to (1) assign a fixed priority to the one or more DMA machines; and (2) assign a programmable priority to the processing unit. The programmable priority of the processing unit allows priority allocation between the one or more DMA machines and the processing unit to be adjusted dynamically.

[0011] Numerous other aspects are provided in accordance with these other aspects of the invention.

[0012] Other features and aspects of the present invention will become more fully apparent from the following detailed description, the appended claims and the accompanying drawings.

Brief Description of Drawings

[0013] FIG. 1 is a block diagram of an integrated circuit (IC) for allocating memory bandwidth in accordance with an embodiment of the present invention.

[0014] FIG. 2 is a block diagram of a novel dynamic priority allocation circuit included in the IC for allocating memory bandwidth shown in FIG. 1 in accordance with an embodiment of the present invention.

[0015] FIG. 3 is a table of exemplary priority values that may be assigned to a processor and a plurality of DMA channels using the dynamic priority allocation circuit in accordance with an embodiment of the present invention.

[0016] FIG. 4 illustrates an exemplary method for allocating memory bandwidth in accordance with an embodiment of the present invention.

[0017] FIG. 5 illustrates a method of allocating memory bandwidth during batch processing in accordance with an embodiment of the present invention.

Detailed Description

[0018] FIG. 1 is a block diagram of an integrated circuit (IC) 100, such as a system on a chip (SOC), for allocating memory bandwidth in accordance with an embodiment of the

present invention. The IC 100 may include a processing unit, such as a processor 102, which may include a subsystem 104 (e.g., internal memory). The internal memory 104 may store data required by instructions to be executed by the processor 102. The internal memory may include a group of instruction memories (IMEMs) and a group of data memories (DMEMs) that may serve as a data resource. The processor 102 may read data and/or execute instructions from the internal memory 104. Although only one processor 102 is shown in the IC 100, it will be understood that the IC 100 may include a plurality of processors 102, each of which includes a subsystem 104.

[0019] The processor 102 may be coupled to one or more devices, such as direct memory access (DMA) devices or machines 106-110 via a bus 112. A DMA device 106-110 may retrieve data required by the processor 102 to execute an instruction and/or move the data required by the processor 102 to the internal memory 104 corresponding to the processor 102.

[0020] The bus 112 (e.g., a processor local bus (PLB) or another suitable bus) provides communication between the DMA devices 106-110, processor 102, and/or the internal memory 104. More specifically, the bus 112 may be cou-

pled to an arbiter 114 (e.g., an arbitration unit) for arbitrating bus access between a plurality of devices. When a device needs access to the bus 112, an interrupt signal may be asserted by the device for notifying the IC 100 that the device is requesting access to the bus 112. The bus 112 may include a plurality of interrupt lines for receiving interrupt signals. Each of the DMA devices 106-110 may be coupled (e.g., hard-wired) to one of the plurality of interrupt lines (e.g., lines $I0_{in}$-$I7_{in}$), respectively, and may assert an interrupt signal on the interrupt line. The arbiter 114 will grant a device access to the bus 112 based on the interrupt signal. If more than one device needs access to the bus 112, the arbiter 114 may grant bus access to one of the devices based on priorities assigned to the interrupt lines from which interrupt signals are received. For example, the arbiter 114 may grant bus access to a device coupled to interrupt line $I0_{in}$ before granting bus access to a device coupled to interrupt line $I7_{in}$ thereby assigning a fixed priority to the devices (e.g., DMA devices 106-110).

[0021] The IC 100 may include a dynamic priority allocation circuit 116, which may be coupled to the arbiter 114, for allocating fixed priorities to the DMA devices 106-110 and

a programmable priority to the processor 102. The programmable priority of the processor allows a priority allocation between the DMA devices 106-110 and the processor 102 to be adjusted dynamically. The dynamic priority allocation circuit 116 may include standard logic and/or may be implemented in an application specific integrated circuit (ASIC) or as a programmable logic circuit. The details of the dynamic priority allocation circuit 116 are discussed below with reference to FIG. 2.

[0022] The IC 100 may include a device, such as a microcontroller 118, coupled to the bus 112. The microcontroller 118 may change a priority assigned to a device included in the IC 100. For example, if the microcontroller 118 receives a command from an operating system of the IC 100 to change the priority of the processor 102, the microcontroller 118 may change the priority assigned to the processor 102. The priority assigned to a task is based on the priority assigned to the device (e.g., processor) from which the task is received. Therefore by changing the priority assigned to the processor 102, the microcontroller 118 may change the priority of a task received from the processor 102. Based on the priority assigned to the task, the microcontroller 118 may allocate memory bandwidth

to the task.

[0023] FIG. 2 is a block diagram of a novel dynamic priority allocation circuit 116 included in the IC 100 shown in FIG. 1 in accordance with an embodiment of the present invention. The dynamic priority allocation circuit 116 may include a first plurality of AND-logic 202-206 coupled to a plurality of interrupt lines $I0_{in}$-$I7_{in}$ included in the bus 112 and an interrupt register 208. More specifically, each of the first plurality of AND-logic 202-206 may be coupled to and receive input from a corresponding interrupt line $I0_{in}$-$I7_{in}$ and a Tx Complete signal, which may indicate whether a DMA device 106-110 is accessing the bus 112. For example, Tx Complete may be of a high logic state when a DMA device 106-110 is accessing the bus 112. Alternatively, Tx Complete may be of a low logic state when no DMA devices 106-110 are accessing the bus 112. Each of the first plurality of AND-logic 202-206 may receive a corresponding one of a plurality of select signals sel 0-sel 7 as inputs. Each of the select signals sel 0-sel 7 serves to activate the AND-logic to which it corresponds. More specifically, if a select signal sel 0-sel 7 coupled to corresponding AND-logic 202-206 is asserted, the AND-logic 202-206 performs a logic AND operation on the sig-

nals provided by the corresponding interrupt line $I0_{in}$-$I7_{in}$ and the Tx Complete line, and outputs the resulting signal for storing in the interrupt register 208. Alternatively, if the select signal sel 0-sel 7 corresponding to AND-logic 202-206 is not asserted, the AND-logic 202-206 may not perform the logic AND operation and may not output a signal to the interrupt register 208. In this manner, the IC 100 for allocating memory bandwidth may perform a logic AND operation and store signals received from an interrupt line $I0_{in}$-$I7_{in}$ to which a device (e.g., DMA device 106-110) is connected in the interrupt register 208. For example, if the IC 100 for allocating memory bandwidth includes eight interrupt lines $I0_{in}$-$I7_{in}$ that may be used for DMA devices, but only includes four DMA devices 106-110, select signals sel 0-sel 3 may be asserted for supporting the four DMA devices 106-110. The remaining select signals sel 4-sel 7 are not asserted. Therefore, the IC 100 may create a DMA channel for moving data corresponding to each DMA device 106-110 included in the IC 100. Each of the DMA channels may be assigned a fixed priority based on the interrupt line coupled to the DMA device to which the DMA channel corresponds. Other numbers of DMA devices 106-110 and/or interrupt lines

$I0_{in}$-$I7_{in}$ may be included in the IC 100 for allocating memory bandwidth.

[0024] The dynamic priority allocation circuit 116 may include a second plurality of AND-logic 210-214 coupled to the interrupt register 208 and a plurality of mask signals MSK0-MSK7. The second plurality of AND-logic 210-214 may output one or more signals stored in the interrupt register 208. More specifically, each of the second plurality of AND-logic 210-214 may be coupled to and receive as input a corresponding signal output by the interrupt register 208 and a corresponding mask signal MSK0-MSK7. Each of the second plurality of AND-logic 210-214 performs a logic AND operation on the corresponding signal received from the interrupt register 208 and the corresponding mask signal MSK0-MSK7, and outputs a resulting signal, which serves as a modified interrupt signal $I0_{out}$-$I7_{out}$. For example, if the mask signal MSK0-MSK7 input to one of the second plurality of AND-logic 210-214 is asserted, the corresponding signal received from the interrupt register 208 and input by the one of the second plurality of AND-logic 210-214 may not be outputted by the AND-logic 210-214 as the modified interrupt signal $I0_{out}$-$I7_{out}$ and therefore, is masked.

[0025] The dynamic priority allocation circuit 116 may include a CPU priority register 216 for storing a programmable priority assigned to the processor 102. The CPU priority register 216 may store a five bit count representing the priority assigned to the processor 102. Other numbers of bits may be stored in the CPU priority register 216. In one embodiment, the dynamic priority allocation circuit 116 may be used for masking signals, which are based on interrupt signals received from DMA devices 106-110 of a lower priority than the processor priority (e.g., the value stored in the CPU priority register 216).

[0026] FIG. 3 is a table 300 of exemplary priority values that may be assigned to the processor 102 (e.g., CPU) and a plurality of DMA channels DMA Channel 0-7 using the dynamic priority allocation circuit 116 in accordance with an embodiment of the present invention. DMA Channels 0-7 may be assigned or allocated priority values of 3, 5, 7, 9, 11, 13, 15, and 17, respectively, where a smaller priority value indicates a higher priority. The priority assigned to each DMA channel DMA Channel 0-7 is fixed. It is based on a priority of a DMA device 106-110 to which the DMA channel corresponds. As stated above, the priority assigned to each of the DMA devices 106-110 is fixed and

based on the priority of the interrupt line to which the DMA device 106-110 is coupled (e.g., hard-wired). The priorities that may be assigned to the processor 102 may be interleaved with the priorities assigned to the DMA channels DMA Channel 0-7, respectively. For example, the processor 102 may be assigned a programmable priority value of 0, 2, 4, 6, 8, 10, 12, 14 or 16, where a lower priority value indicates a higher priority. Different values may be used for the priority values of the DMA channels and/or the processor 102. A larger or smaller number of values may be used. The processor priority 0 may be used when the processor is reset.

[0027] As stated above, the value of the programmable priority assigned to the processor 102 may be stored in the CPU priority register 216. By changing the value stored in the CPU priority register 216, the priority of the processor may be adjusted to be higher or lower than one or more of the plurality of DMA channels DMA Channel 0-7. The priority of a task to be performed by a device (e.g., processor 102 or DMA device 106-110) may be the priority assigned to the device; and tasks of a higher priority may be executed by the IC 100 for allocating memory bandwidth before tasks of a lower priority. Therefore, changing

the priority assigned to the processor 102 may affect the order in which tasks are performed by the IC 100.

[0028] A priority value (e.g., a fixed priority value) may be assigned to a slave device, such as a device (e.g., a microcontroller 118) used for changing the value stored in the CPU priority register 216. The priority value may be assigned such that the priority of the slave device is higher than the priority of the DMA channels DMA Channel 0-7 and most of the priorities that may be assigned to the processor 102. In this manner, the slave device may be granted access to the bus 112 in response to an interrupt signal from the slave device before DMA devices 106-110 which have also asserted interrupt signals. Similarly, the slave device may be granted access to the bus 112 in response to an interrupt signal from the slave device before the processor 102, which has also asserted an interrupt signal, if the processor 102 is assigned a priority lower than the slave device. Such a priority scheme allows the slave device to access the bus 112 and subsequently adjust the priority assigned to the processor 102 without waiting for DMA device tasks and most processor tasks to complete.

[0029] The operation of the IC 100 for allocating memory band-

width is now described with reference to FIGS. 1-3 and with reference to FIG. 4 which illustrates an exemplary method for allocating memory bandwidth. With reference to FIG. 4, in step 402, the method 400 begins. In step 404, a fixed priority is assigned to one or more direct memory access (DMA) machines or devices. As stated each of the one or more DMA devices 106-110 may be coupled (e.g., hard-wired) to an interrupt line $I0_{in}$-$I7_{in}$ included in the bus 112; and the interrupt lines $I0_{in}$-$I7_{in}$ are prioritized such that the arbiter 114 may grant bus access to a device coupled to a first interrupt line (e.g., $I0_{in}$) before a device coupled to a second interrupt line $I7_{in}$. More specifically, each of the interrupt lines may be of a different priority. The priority assigned to the one or more DMA devices 106-110 may be based on the interrupt line $I0_{in}$-$I7_{in}$ to which the DMA device 106-110 is connected. Because the one or more DMA devices 106-110 are hard-wired to the IC 100, and therefore, to one or more interrupt lines $I0_{in}$-$I7_{in}$, the priority assigned to the one or more DMA devices 106-110 is fixed. For example, if the IC 100 for allocating memory bandwidth includes eight DMA devices 106-110, the DMA devices 106-110 may be assigned priorities of 3, 5, 7, 9, 11, 13, 15, and 17, re-

spectively. The priority assigned to a DMA channel corresponding to a DMA device 106-110 may be the priority assigned to the DMA device 106-110.

[0030] In step 406, a programmable priority may be assigned to a processing unit (e.g., processor 102). A device included in the IC 100 for allocating memory bandwidth may assign one of a plurality of priority values to the processor 102. For example, the microcontroller 118 may receive a command from the operating system of the IC 100 to assign the processor 102 a priority value. In response, the microcontroller 118 may write a count corresponding to the priority value in the CPU priority register 216 included in the dynamic priority allocation circuit 116. More specifically, the microcontroller 118 may be coupled to an interrupt line which causes the microcontroller 118 to have a priority of 1. The microcontroller 118 may assert an interrupt signal on the interrupt line to gain access to the bus 112. Once the microcontroller 118 is granted access to the bus 112, the microcontroller 118 may write the count corresponding to the priority value to the CPU priority register 216. In this manner, as stated, the processor 102 may be assigned a programmable priority of 0, 2, 4, 6, 8, 10, 12, 14, or 16; and the priorities that may be assigned

the processor 102 are interleaved with the priorities assigned to the one or more DMA devices 106-110. Therefore, the programmable priority of the processor 102 allows a priority allocation between the one or more DMA devices 106-110 and the processor 102 to be adjusted dynamically.

[0031] In step 408, memory bandwidth is allocated to a task to be performed by one of the one or more DMA machines or devices (e.g., using a DMA channel) and the processing unit (e.g., processor) of the highest priority. An application running on the IC 100 may cause one or more of the DMA devices 106-110 and the processor 102 to perform tasks. Because the application runs with a real-time budget, one or more of the tasks may need to be completed within an allotted time. For example, a DMA device 106-110 may need to move data, which is required by an instruction to be executed by the processor 102, to the internal memory 104 of the processor 102 before the instruction is executed. Similarly, the processor 102 may need to execute an instruction or read data from the internal memory 104 within a given time frame. If either one or more of the DMA devices 106-110 or the processor 102 does not complete a task within a required time

frame, the IC 100 fails and the application is non-operative. A task to be performed by a DMA device 106-110 may be assigned the priority of the DMA device. Similarly, a task to be performed by the processor 102 may be assigned the priority of the processor 102.

[0032] The one or more DMA devices 106-110 and/or the processor 102 may need to access the internal memory 104 via the bus 112 to perform required tasks within an allotted time. Because the memory 104 (e.g., internal memory) may be of a limited bandwidth and only one device may be granted access to the bus 112 at a time, memory bandwidth may be allocated to the highest-priority task to be performed by the DMA device 106-110 or the processor 102. In this manner, the IC 100 for allocating memory bandwidth may grant bus access and subsequently allocate memory bandwidth to the most critical task (e.g., highest-priority task) before other tasks which require bus access and memory bandwidth.

[0033] In one embodiment, access to the memory bandwidth is delayed for the one or more DMA devices 106-110 assigned a priority lower than the processor 102. Therefore, tasks to be performed by the one or more DMA devices 106-110 assigned a priority lower than the processor 102

will not be allocated memory bandwidth until tasks from higher-priority devices are allocated memory bandwidth. More specifically, interrupt signals received from the DMA devices 106-110 of a lower priority than the processor 102 may be masked, as described above, while discussing the dynamic priority allocation circuit 116 of FIG. 2. By masking the interrupt signals, the dynamic priority allocation circuit 116, prevents interrupt signals from the one or more DMA devices 106-110 of a priority lower than the processor 102 from reaching the bus 112. Consequently, such DMA devices 106-110 may not access the memory 104, and therefore, may not be allocated memory bandwidth.

[0034] In step 410, the priority of the processing unit may be adjusted. More specifically, the priority assigned to the processor 102 may be adjusted by changing the value stored in the CPU priority register 216. For example, in response to a command from the operating system, the microcontroller 118 may assert an interrupt signal for accessing the CPU priority register 216 via the bus 112 and write a new value (e.g., a count) to the CPU priority register 216 thereby adjusting the priority of the processor 102. The priority assigned to the processor 102 may be increased

such that the processor priority is higher than a greater number of DMA devices 106-110 or decreased such that the processor priority is lower than a greater number of the DMA devices 106-110.

[0035] For example, while an application is running on the IC 100 for allocating memory bandwidth, a task (e.g., a critical task) to be performed by the processor 102 may be in jeopardy of not completing within an allotted time, which would result in failure of the IC 100. To avoid such a failure, the priority assigned to the processor 102, and therefore the task to be performed by the processor 102, may be changed (e.g., increased). In this manner, the processor 102 may be granted access to the bus 112 and allocated memory bandwidth before a greater number (perhaps all) of the DMA devices 106-110, which may allow the processor task to complete within the allotted time.

[0036] Alternatively, the processor 102 may not have to perform tasks of a high priority, but one or more tasks that must be performed by a DMA device 106-110 using a corresponding DMA channel may be in jeopardy of not completing within an allotted time, which may result in an IC 100 failure. The dynamic priority allocation circuit 116

may decrease the priority of the processor 102 below the priority of the DMA devices 106-110, and therefore, tasks (e.g., critical tasks) corresponding to such DMA devices 106-110 are allocated memory bandwidth before other DMA devices 106-110 and the processor 102. Consequently, the critical tasks may be completed within the allotted time.

[0037] In step 412, the method 400 ends. Through the use of the method of FIG. 4, memory bandwidth may be allocated to devices, such as one or more DMA devices 106-110 and a processor 102, included in an IC 100. Tasks are performed by the devices based on priorities assigned to the devices (and therefore assigned to the tasks), respectively, such that each task may be performed within an allotted time. The priorities assigned to the one or more DMA devices 106-110 may be fixed and the priority assigned to the processor 102 may be programmable, which allows the priority allocation between the one or more DMA devices 106-110 (and therefore the DMA channels used by the one or more DMA devices 106-110) and the processor 102 to be adjusted dynamically.

[0038] The operation of the IC 100 for allocating memory bandwidth is now described with reference to FIGS. 1-4, and

with reference to FIG. 5 which illustrates a method 500 of allocating memory bandwidth during batch processing in accordance with an embodiment of the present invention. More specifically, FIG. 5 illustrates a first batch processing Batch 1 that must be performed at a first rate (e.g., between time t_1 and t_3, between time t_3 and t_4, etc.). During each allotted time (e.g., between time t_1 and t_3) of the Batch 1 processing, a data movement task Data Movement 1 must be performed by a first DMA device using a corresponding DMA channel (e.g., DMA Channel 0) and a data execution task Data Execution 1 must be performed by a processor. Similarly, FIG. 5 illustrates a second batch processing Batch 2 that must be performed at a second rate (e.g., between time t_2 and t_5). During the allotted time, a data movement task Data Movement 2 must be performed by a second DMA device using a corresponding DMA channel (e.g., DMA Channel 1) and a data execution task Data Execution 2 must be performed by the processor.

[0039] As stated above while describing step 404, a fixed priority may be assigned to each of the first and second DMA devices. Tasks to be performed by a DMA device and a DMA channel used for performing the tasks are assigned the same priority as the DMA device to which they corre-

spond. For example, the first DMA device may be assigned a priority of 3 and the second DMA device may be assigned a priority of 5. Therefore, the priority of the Data Movement 1 task and DMA Channel 0 is 3, and the priority of the Data Movement 2 task and DMA Channel 1 is 5.

[0040] As stated above while describing step 406, a programmable priority may be assigned to the processor 102. For example, the processor 102 may be assigned a priority of 4. Because tasks to be performed by the processor 102 are assigned the same priority as the processor 102, the priority of Data Execution 1 and Data Execution 2 is 4.

[0041] As stated above while describing step 408, memory bandwidth is allocated to the task to be performed by one of the one or more DMA devices 106-110 and the processor 102 of the highest priority. In the above example, at time t_1, the dynamic priority allocation circuit 116 will allocate memory bandwidth to Data Movement 1 (to be performed by the first DMA device), which is of the highest priority (e.g., 3). Once Data Movement 1 is completed, the dynamic priority allocation circuit 116 will allocate memory bandwidth to Data Execution 1 (to be performed by the processor), which is of the highest priority (e.g., 4). It

should be noted that although Data Execution 1 and Data Execution 2 are to be performed by the processor 102, because Batch 1 processing is of a higher priority than Batch 2 processing, Data Execution 1 is performed before Data Execution 2. Therefore, the Batch 1 processing is completed within the allotted time (e.g., between time t_1 and t_3).

[0042] As stated above while describing step 410, the priority of the processor 102 may be adjusted. More specifically, after Data Execution 1 is performed by the processor 102 (e.g., at time t_2), the dynamic priority allocation circuit 116 may lower the priority of the processor 102 to 6. Therefore, at time t_2, the dynamic priority allocation circuit 116 will allocate memory bandwidth to Data Movement 2 (to be performed by the second DMA device), which is of the highest priority (e.g., 5). However, while performing Data Movement 2, at time t_3, the first DMA device asserts an interrupt signal on an interrupt line $I0_{in}$ - $I7_{in}$ for access to the bus 112 to perform Data Movement 1. Because the first DMA device is of a higher priority (e.g., 3) than the second DMA device (e.g., 5), which is performing Data Movement 2, Data Movement 2 is interrupted and Data Movement 1 is performed. In this manner

data required by the processor 102 to execute an instruction (e.g., Data Execution 1) may be moved to the memory 104 (by performing Data Movement 1) before executing the instruction. Data Movement 2 will not resume until Data Movement 2 is the highest-priority task.

[0043] At time t_3, the priority of the processor 102 is increased to 4. Consequently, after Data Movement 1 is completed, Data Execution 1 may be performed, because it may be of the highest priority. In this manner, the Batch 1 processing is performed again within the allotted time (e.g., between time t_3 and t_4). After Data Execution 1 is performed by the processor 102 (e.g., between times t_3 and t_4), the dynamic priority allocation circuit 116 may lower the priority of the processor 102 to 6. At this time Data Movement 2, which is of the highest priority (e.g., 5) resumes. Thereafter, the batch processing continues in a similar manner. Using the present methods, Batch 1 processing is completed within the allotted time (e.g., between time t_1 and t_3, between time t_3 and t_4, etc.) and Batch 2 processing is completed within the allotted time (e.g., between time t_2 and t_5).

[0044] The foregoing description discloses only exemplary embodiments of the invention. Modifications of the above

disclosed embodiments of the present invention which fall within the scope of the invention will be readily apparent to those of ordinary skill in the art. For instance, although in the above embodiments a fixed priority was assigned to DMA devices 106-110, in other embodiments, a fixed priority may be assigned to other devices as well. Further, although in the above embodiments, a subsystem 104 corresponds to a processor 102, in other embodiments, a plurality of processors may share and/or access the same subsystem. In such embodiments, the priorities that may be assigned to each of the plurality of processors may be interleaved with the remaining processors and devices (e.g., DMA devices 106-110) included in the IC 100. Further, in other embodiments, multiple DMA devices 106-110 may be used for accessing one or more subsystems, where a subsystem is included for each of a plurality of processors 102. Although in the above embodiments, a low priority value indicates a high priority, in other embodiments, the reverse may be true. In other embodiments, additional fixed and/or programmable priorities corresponding to a device may be interleaved with the priority values described above.

[0045] Further, although in one or more embodiments, a single

DMA channel corresponds to a DMA device, in other embodiments, one or more DMA channels may correspond to a DMA device. While the dynamic priority allocation circuit 116 is shown to include a first and second plurality of AND-logic, in other embodiments, different logic may be used.

[0046] In one or more embodiments of the present invention, a timer (e.g., an external timer) may be used for determining whether a task to be performed by the processor 102 is in jeopardy of not being completed within an allotted time. When a time-out condition may occur, an external trigger may be adapted to override (e.g., via an interrupt) a programmed priority of the processor 102 and force the priority of the processor 102 to a predefined hardware level (e.g., the highest priority). Such an interrupt may be serviced with a priority different than the processor 102 and the one or more DMA devices 106-110, and therefore, gives an IC designer full control of the priorities assigned to the devices included in the IC 100.

[0047] In one or more embodiments of the present invention, a device of the highest priority is allocated memory bandwidth by stalling devices of lower priorities until the highest-priority device no longer needs the memory band-

width. For example, clock-gating techniques may be used for stopping the processor 102 (e.g., putting the processor 102 in sleep mode). Alternatively, an acknowledgement used by a bus protocol may not be transmitted while the highest-priority device is accessing the memory bandwidth.

[0048] In one or more embodiments of the present invention, the operating system (OS) included in the IC 100 is provided with the memory bandwidth requirements of each task to be performed by the IC 100. The OS may be programmed for managing the priority values included in the dynamic priority allocation circuit 116. The OS may program monitors for ensuring tasks are completed within an allotted time. For example, if a monitor is triggered, a hardware interrupt may be used for assigning the processor 102 the highest priority of the system. A signal may be sent to the processor 102, and in response to the signal the processor 102 may execute code for determining whether the task currently performed by the processor 102 has enough memory bandwidth. If not, the priority of the processor 102 will remain at the highest priority level until the processor 102 completes the task. Thereafter, the priority of the processor 102 may be lowered. Alternatively,

if the processor 102 has access to sufficient memory bandwidth for completing the current task, a next external monitor may be set, and the hardware interrupt is exited. Once the hardware interrupt is exited, the priority previously-assigned to the processor 102 is restored and interrupt signals that were masked (by the dynamic priority allocation circuit 116) before the hardware interrupt will be masked again.

[0049] Accordingly, while the present invention has been disclosed in connection with exemplary embodiments thereof, it should be understood that other embodiments may fall within the spirit and scope of the invention as defined by the following claims.